1. The ten best-selling video games

Photo by Dan Schleusser (https://unsplash.com/@retromoon) on Unsplash (https://unsplash.com).

Video games are big business: the global gaming market is projected to be worth more than $300 billion by 2027 according to Mordor Intelligence (https://www.mordorintelligence.com/industry-reports/global-gaming-market). With so much money at stake, the major game publishers are hugely incentivized to create the next big hit. But are games getting better, or has the golden age of video games already passed?

In this project, we'll explore the top 400 best-selling video games created between 1977 and 2020. We'll compare a dataset on game sales with critic and user reviews to determine whether or not video games have improved as the gaming market has grown.

Our database contains two tables. We've limited each table to 400 rows for this project, but you can find the complete dataset with over 13,000 games on Kaggle (https://www.kaggle.com/holmjason2/videogamedata).

`game_sales`

column	type	meaning
game	varchar	Name of the video game
platform	varchar	Gaming platform
publisher	varchar	Game publisher
developer	varchar	Game developer
games_sold	float	Number of copies sold (millions)

column	type	meaning
year	int	Release year

reviews

column	type	meaning
game	varchar	Name of the video game
critic_score	float	Critic score according to Metacritic
user_score	float	User score according to Metacritic

In [19]:

```
%%sql
postgresql:///games

select *
from game_sales
order by games_sold desc
limit 10
```

10 rows affected.

Out[19]:

game	platform	publisher	developer	games_sold	year
Wii Sports for Wii	Wii	Nintendo	Nintendo EAD	82.90	2006
Super Mario Bros. for NES	NES	Nintendo	Nintendo EAD	40.24	1985
Counter-Strike: Global Offensive for PC	PC	Valve	Valve Corporation	40.00	2012
Mario Kart Wii for Wii	Wii	Nintendo	Nintendo EAD	37.32	2008
PLAYERUNKNOWN'S BATTLEGROUNDS for PC	PC	PUBG Corporation	PUBG Corporation	36.60	2017
Minecraft for PC	PC	Mojang	Mojang AB	33.15	2010
Wii Sports Resort for Wii	Wii	Nintendo	Nintendo EAD	33.13	2009
Pokemon Red / Green / Blue Version for GB	GB	Nintendo	Game Freak	31.38	1998
New Super Mario Bros. for DS	DS	Nintendo	Nintendo EAD	30.80	2006
New Super Mario Bros. Wii for Wii	Wii	Nintendo	Nintendo EAD	30.30	2009

2. Missing review scores

Wow, the best-selling video games were released between 1985 to 2017! That's quite a range; we'll have to use data from the `reviews` table to gain more insight on the best years for video games.

First, it's important to explore the limitations of our database. One big shortcoming is that there is not any `reviews` data for some of the games on the `game_sales` table.

In [21]:

```sql
%%sql

select count(*)
from game_sales
left join reviews
on game_sales.game = reviews.game
where critic_score is null and user_score is null
```

 * postgresql:///games
1 rows affected.

Out[21]:

count

　31

3. Years that video game critics loved

It looks like a little less than ten percent of the games on the `game_sales` table don't have any reviews data. That's a small enough percentage that we can continue our exploration, but the missing reviews data is a good thing to keep in mind as we move on to evaluating results from more sophisticated queries.

There are lots of ways to measure the best years for video games! Let's start with what the critics think.

In [23]:

```sql
%%sql


select g.year, round(avg(r.critic_score),2) as avg_critic_score
from game_sales as g
left join reviews as r
on g.game = r.game
group by g.year
order by avg_critic_score desc
limit 10
```

 * postgresql:///games
10 rows affected.

Out[23]:

year	avg_critic_score
1990	9.80
1992	9.67
1998	9.32
2020	9.20
1993	9.10
1995	9.07
2004	9.03
1982	9.00
2002	8.99
1999	8.93

4. Was 1982 really that great?

The range of great years according to critic reviews goes from 1982 until 2020: we are no closer to finding the golden age of video games!

Hang on, though. Some of those `avg_critic_score` values look like suspiciously round numbers for averages. The value for 1982 looks especially fishy. Maybe there weren't a lot of video games in our dataset that were released in certain years.

Let's update our query and find out whether 1982 really was such a great year for video games.

In [25]:

```sql
%%sql


select g.year, count(r.game) as num_games, round(avg(r.critic_score),2) as avg_critic_scc
from game_sales as g
left join reviews as r
on g.game = r.game
group by g.year
having count(r.game) > 4
order by avg_critic_score desc
limit 10
```

```
 * postgresql:///games
10 rows affected.
```

Out[25]:

year	num_games	avg_critic_score
1998	10	9.32
2004	11	9.03
2002	9	8.99
1999	11	8.93
2001	13	8.82
2011	26	8.76
2016	13	8.67
2013	18	8.66
2008	20	8.63
2017	13	8.62

5. Years that dropped off the critics' favorites list

That looks better! The `num_games` column convinces us that our new list of the critics' top games reflects years that had quite a few well-reviewed games rather than just one or two hits. But which years dropped off the list due to having four or fewer reviewed games? Let's identify them so that someday we can track down more game reviews for those years and determine whether they might rightfully be considered as excellent years for video game releases!

It's time to brush off your set theory skills. To get started, we've created tables with the results of our previous two queries:

top_critic_years

column	type	meaning
year	int	Year of video game release
avg_critic_score	float	Average of all critic scores for games released in that year

top_critic_years_more_than_four_games

column	type	meaning
year	int	Year of video game release

In [27]:

```sql
%%sql

select year, avg_critic_score
from top_critic_years
group by year
except
select year, avg_critic_score
from top_critic_years_more_than_four_games
group by year
order by avg_critic_score desc
```

 * postgresql:///games
6 rows affected.

Out[27]:

year	avg_critic_score
1990	9.80
1992	9.67
2020	9.20
1993	9.10
1995	9.07
1982	9.00

6. Years video game players loved

Based on our work in the task above, it looks like the early 1990s might merit consideration as the golden age of video games based on `critic_score` alone, but we'd need to gather more games and reviews data to do further analysis.

Let's move on to looking at the opinions of another important group of people: players! To begin, let's create a query very similar to the one we used in Task Four, except this one will look at `user_score` averages by year rather than `critic_score` averages.

In [29]:

```sql
%%sql


select g.year, round(avg(r.user_score),2) as avg_user_score, count(r.game) as num_games
from game_sales as g
left join reviews as r
on g.game = r.game
group by g.year
having count(r.game) > 4
order by avg_user_score desc
limit 10
```

 * postgresql:///games
10 rows affected.

Out[29]:

year	avg_user_score	num_games
1997	9.50	8
1998	9.40	10
2010	9.24	23
2009	9.18	20
2008	9.03	20
1996	9.00	5
2005	8.95	13
2006	8.95	16
2000	8.80	8
2002	8.80	9

7. Years that both players and critics loved

Alright, we've got a list of the top ten years according to both critic reviews and user reviews. Are there any years that showed up on both tables? If so, those years would certainly be excellent ones!

Recall that we have access to the `top_critic_years_more_than_four_games` table, which stores the results of our top critic years query from Task 4:

`top_critic_years_more_than_four_games`

column	type	meaning
year	int	Year of video game release
num_games	int	Count of the number of video games released in that year
avg_critic_score	float	Average of all critic scores for games released in that year

We've also saved the results of our top user years query from the previous task into a table:

top_user_years_more_than_four_games

column	type	meaning
year	int	Year of video game release

In [31]:

```sql
%%sql

select year
from top_critic_years_more_than_four_games
group by year
intersect
select year
from top_user_years_more_than_four_games
group by year
```

 * postgresql:///games
3 rows affected.

Out[31]:

year
1998
2008
2002

8. Sales in the best video game years

Looks like we've got three years that both users and critics agreed were in the top ten! There are many other ways of measuring what the best years for video games are, but let's stick with these years for now. We know that critics and players liked these years, but what about video game makers? Were sales good? Let's find out.

This time, we haven't saved the results from the previous task in a table for you. Instead, we'll use the query from the previous task as a subquery in this one! This is a great skill to have, as we don't always have write permissions on the database we are querying.

In [33]:

```sql
%%sql

select year, sum(games_sold) as total_games_sold
from game_sales
group by year
having year in (select year
from top_critic_years_more_than_four_games
group by year
intersect
select year
from top_user_years_more_than_four_games
group by year)
order by year desc
```

 * postgresql:///games
3 rows affected.

Out[33]:

year	total_games_sold
2008	175.07
2002	58.67
1998	101.52

```sql
%%sql
```